---------------------- CEN 4072/6070 Software Testing & Verification ----------------------

Exam 2 – Spring 2015 – Solution Notes

1. a. would not (t could = -5 and z could = 1 initially)
   b. would not (the value of z could also be 17 when S terminates)
   c. would not (weak correctness does not require that P => termination)
   d. would  ($y<z \Rightarrow y\neq z+1$)
   e. would not (The fact that t=z on termination of S when |t|>2 initially does not imply that y cannot equal z+1 on termination of S when z<0 initially.  Can you identify a program S for which this would be the case?)
   f. would (This implies that for z=-4, -3, -2, and -1 initially, S WILL terminate in a state that is inconsistent with the given post-condition.)
   g. would not (Here it may be the case that S does not terminate when z=-4, -3, -2, and -1 initially.)
   h. would not (This does not imply that y cannot equal z+1 on termination of S when z<0 initially.  See notes for part e.)

2. a. false ($x>0 \not\Rightarrow x=5$)
   b. true (IF the program terminates, it will terminate in state Q.)
   c. true
   d. true (initialization and preservation hold)
   e. true (initialization, preservation, and finalization hold)
   f. true
   g. false (See problem set 5.)
   h. true
   i. false (This would only be true if P => k.)
   j. true

3. a.

$$\frac{P \Rightarrow wp(S,Q)}{\{P\}\ S\ \{Q\}\ \text{strongly}}$$

   b. $H_0$: $(x=0 \land y=17)$
      $H_1$: $(x=1 \land y=14)$
      $H_2$: $(x=2 \land y=11)$
      $H_k$: $(x=k \land y=17-3k)$
      Closed form expression of wp: $\mathbf{(x\geq 0 \land y=17-3x)}$
   c. Clearly, the instantiated antecedent of the ROI, $(x=5 \land y=2)\Rightarrow(x\geq 0 \land y=17-3x)$, holds, and we therefore conclude that the assertion holds.

4. g

5.

| | *P1* | *P2* |
|---|---|---|
| *f1* | S | N |
| *f2* | N | C |

p1 = (x>2 -> x,z := 2,z(x-1)!  | true -> x,z := x-1,z(x-1)))
p2 = (x>1 -> x,z := 1,z(x-1)!)

6. a. valid
   b. invalid (true ≠> Q for every Q)
   c. invalid (the antecedent does not imply termination if P holds)
   d. valid
   e. invalid (antecedents do not guarantee that either I or P will hold if S executes more than once)
   f. valid
   g. valid
   h. valid
   i. valid
   j. invalid (the antecedent implies that S, if executed, will terminate, but it does NOT imply that while b do S will terminate)

7. **INITIALIZATION: Does {P} s {I}?**

   P: $\{n \geq -17 \land t=1 \land k=0\}$
   $t := 2*t$
   $\{n \geq -17 \land t=2 \land k=0\}$
   $k := k+1$
   $\{n \geq -17 \land t=2 \land k=1\} \Rightarrow t=2^{\mathbf{k}}$ **=** I

   Therefore {P} s {I}. √

   **PRESERVATION: Does {I ∧ ¬b} s {I}?**

   I ∧ ¬b: $\{ t=2^{\mathbf{k}} \land k \neq n \}$
   $t := 2*t$
   $\{ t=2^{\mathbf{k+1}} \land k \neq n \}$
   $k := k+1$
   $\{ t=2^{\mathbf{(k-1)+1}} \land k-1 \neq 0 \}$
   $\Leftrightarrow$
   $\{ t=2^{\mathbf{k}} \land k-1 \neq 0 \} \Rightarrow$ I √

   **FINALIZATION: Does (I ∧ b) ⇒ Q?**

   (I ∧ b): $(t=2^{\mathbf{k}} \land k=n) \Rightarrow t=2^{\mathbf{n}}$ = Q √

8. **Does term*(f, H)?***

   We use the Method of Well Founded Sets with measure k to prove *H* will terminate for any initial (integer) values of k and n such that k<n.

   i. the value of k increases by 1 with each execution of the loop body (via k := k+1).

   ii. the value of k is bounded from above when k is initially less than n since when k becomes equal to n (which is constant), the loop must terminate **because** "k=n" (i.e., the loop predicate) becomes true.

iii. the value of k may assume only a finite number of values [($k_0$<n, $k_0$+1, $k_0$+2, …,n)] since it increases by an integral amount (1) with each iteration of the loop body.

Therefore, *H* terminates for any initial value of k<n and we conclude that term*(f, H) holds.*

**Does *(p o g)* ⇒ *(f = g)*?**

[ (k=n) *o* (t,k := 2t,k+1) ] ⇒ ($k_0$=n-1)

(k=n-1) ⇒ ( *f* = (t,k := $t2^{\mathbf{n-(n-1)}}$,k+1)
= **(t,k := 2t,k+1)** )

(k=n-1) ⇒ ( *g* = **(t,k := 2t,k+1)** ).

Therefore, *(p o g)* ⇒ *(f = g)*.

**Does ¬*(p o g)* ⇒ *(f = f o g)*?**

**¬** [ (k=n) *o* (t,k := 2t,k+1) ] ⇒ ($k_0$≠n-1)

Thus, there are 2 cases to consider: $k_0$<n-1 and $k_0$>n-1.
case a:
(k<n-1) ⇒ ( *f* = **(t,k := $t2^{n-k}$,n)** )
(k<n-1) ⇒ ( *f* o *g* = (t,k := $t2^{n-k}$,n) o
(t,k := 2t,k+1)
since ((k<n) o *g*(k<n-1)) = true
= (t,k := $(2t)2^{n-(k+1)}$,n)
= (t,k := $t2^{n-k-1+1}$,n)
= **(t,k := $t2^{n-k}$,n)** )
case b:
(k>n-1) ⇒ ( *f* = **undefined** )
(k>n-1) ⇒ ( *f* o *g* = undefined o *g*
since ((k<n) o *g*(k>n-1)) = false
= **undefined** )
Therefore, ¬*(p o g)* ⇒ *(f = f o g).*

9. a. i. $y_n$ = 0, $z_n$ = $z_0$-3$y_0$
ii. $t(X_n)$ = $X_n$ = (0, $z_0$-3$y_0$)
iii. $q$(X): z-3y = $z_0$-3$y_0$ or z = $z_0$+3(y-$y_0$)

b. One cannot deduce unique values of $y_0$ and $z_0$ from this information, but one CAN deduce that $z_0$-3$y_0$ = -1

c. iii

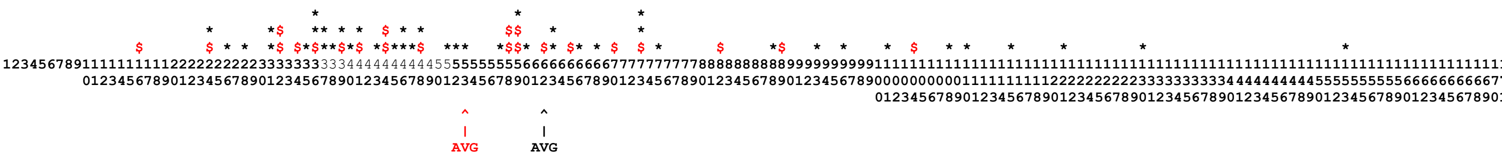
d. No, any program of the form *while b do S* computing *t* that produces intermediate state (5,-7) would necessarily terminate with y=0 and z=-22, while any such program that produces intermediate state (3,8) would necessarily terminate with y=0 and z=-1. (In short, *t*(5,-7) ≠ *t*(3,8), and therefore *q*(5,-7) and *q*(3,8) are inconsistent.) Therefore, there is no input $X_0$ for which the program could produce both intermediate states.

10. false, false, true, true, true

11. false, true, false, false, false

12. (x>1 → x,y := 3,x-1 | x≤1 → x,y := 3, 1-x)

## Histogram of Raw Scores




* - CEN 6070 students
$ - CEN 4072 students